

Data Architecture & Administration

2008 Data Naming Standard

This data element strategy will be used in conjunction with the Data Name Standard and the Previous Data Naming Alternative developed in 1994. While the 1994 alternative is still supported, this alternative will be the preferred approach to assigning data element names, and will be used for all new database modeling and physical data element naming once this alternative is approved.

At no time will one system use both naming alternatives unless sharing linked tables from one system to the other. Existing systems are not required to be re-modeled unless they are being totally redeveloped.

The purpose of a data name strategy is to standardize data element names and to define a standard collection of classwords which group large numbers of attributes into broad categories.

In an effort to simplify the task of understanding the underlying data type of a data column, classwords will be suffixed to column names in a consistent manner. In the past, classword codes (2 Alpha characters following an underscore) were used in an effort to save space when naming columns. This was necessary due to column name length constraints in legacy data management systems. Modern technology does not constrain developers to a short count of characters allowed in a column name. The new standard spells out the entire classword and omits the underscore prefix which used to be added between the column name and the classword code suffix. For example, a column that was called *My_Customer_NM* will now be called *MyCustomerName*. All column names will be Pascal cased (First letter of each word upper case followed by Lower case letters to complete each word)

In some cases a data modeler may want to abbreviate one or more words in a table. For consistency, each occurrence of that word in the database would also be abbreviated.

The agency will maintain a classword table and an approved process will be in place for administering amendments to the table.

It is very important to limit the number of data classwords to prohibit the classwords from becoming cryptic in nature and creating undue complexity in data naming standards. It is as important to populate the list of acceptable classwords with a complete collection such that no classword is compromised nor used in an inappropriate manor. Information in addition to generic data type supplied by the classword will be found in the data dictionary.

The following table contains the list of draft classwords that will be used in this alternative for naming attributes. The **Legacy Codes** are in a separate document and are not part of this alternative.

Note that the Domain column fields identified as Text can be any domain of text data, char, varchar, nvarchar or other. Also numeric is any number domain as int, double and other.

Suffix Classword	Description	Domain	Total Length	Decimal Places
Address	Descriptive text used to denote a place where a person or organization may be communicated with (i.e. PO Box), or a physical location (i.e. Street Address).	Text, nvarchar	50	
Amount	Most values of the number data type expressed as two decimal places which are meant to define currency. For special accounting purposes four digits to the right of the decimal point are acceptable.	Number	15	4
BLOB	A subset of Blobs that are non-imaged data. This includes data that is compressed, zipped, or otherwise encrypted.	Binary	Variable	
Code	A numeric or character value that identifies classifications or categories of a member of a set of like values. A code does not include the description of the code value rather a simple abbreviation that stands for that description.	Number or Text as is required	Integer or Variable not to exceed 9 bytes.	-

Constant	Data which does not change value over time or in different circumstances or uses.	Text, or number	Numeric or a Variable not to exceed 8	
Date	A unit of time expressed in months, days, and years, used to designate a specific 24-hour period.	Date	8	
Datetime	A specific instance of time that includes date and time components.	Dependent on DBMS used.		
Decimal	A numeric representation of data that is not normally considered a quantity and represented in float decimal or numeric notation with or without significant digits to the right of the decimal point.	Number preferred Or Float if no other options is available	Variable	Variable
Description	Data having undefined, freeform, unstructured, or unformatted content and is not an Address or Name.	Text, nvarchar	Variable	
Flag	A bit or series of bits with two stable states. A binary condition permitting only two values (i.e. True/False, Yes/No, Pass/Fail).	Char(1) or bit	1	
Hash	A resulting hash from Secure Hash Algorithm-256 (SHA256) or higher when available.	Text, nvarchar	Variable	
Id	Either a numeric value that implies sequence; or a computer generated serial identifier used to generate primary keys in a database to maintain referential integrity.	Number	9	
Image	A subset of Binary Large Objects, Blobs, that represents a digitized or scanned image or document. (This includes PDF's, bitmaps, jpegs and other image forms and document types.	Binary	Variable	

Name	A word or phrase that constitutes the distinctive designation of a person, place, thing, or concept.	Text, nvarchar	Variable	
ShortName	A word or phrase that constitutes the distinctive designation of a person, place, thing, or concept in a shortened string. Acronyms, codes, etc. will be valid uses of this field.	Text, nvarchar	Variable	
Number	A combination of letters and/or numbers used to uniquely identify an occurrence of something. (i.e. Social Security Number, Vehicle Identification Number). Special characters used as separators would be excluded from occurrences of attributes or fields in this class. Rather, display formats would achieve this effect. For example, Social Security Number would be 9 digits without the 2 '-' separators.	Number or Text as is required	Variable	
Percent	A number that represents the ratio between two values that have the same unit of measure multiplied by 100 (A rate times 100).	Number	5	2
Quantity	A number of non-monetary units expressed in conjunction with a unit of measure.	Number	Dependent on associated Measurement and Unit	
Rate	A quantity or amount measured with respect to another measured quantity or amount (i.e. dollars/hour, miles/gallon, etc.).	Number	9	4
XML	A valid XML Document, which could be XML, XSLT, Schema, or other well-formed XML document type.	XML	Variable	

Guidance on Designing Lookup tables

Lookup tables will be created using one schema. This schema ensures that future lookup data sort requests, item expiration, and acronyms will be natively available from the onset of a project for all lookup tables. Below is a SQL script template for creating lookup tables in the database. Replacing all instances of “*tablename*” in the script below with the name of the lookup table will result in an

execution ready script to add a lookup table to the database. It is recommended that lookup tables be named to include the word 'Type' or 'Lookup' in the name to easily identify them as look-up tables (e.g., TablenameType, TablenameLU, or Tablename_LookUp). The approach should be consistent throughout the database.

```
CREATE TABLE [dbo].[TablenameType] (
    [TablenameTypeId] [int] identity NOT NULL,
    [TablenameTypeDisplayShortName] [varchar](50) NOT NULL,
    [TablenameTypeDisplayName] [varchar](128) NOT NULL,
    [TablenameTypeDescription] [varchar](500) NOT NULL,
    [SortOrderNumber] [int] NOT NULL CONSTRAINT
[DF_tablenameType_SortOrderNumber] DEFAULT ((99)),
    [IsActiveFlag] [bit] NOT NULL CONSTRAINT
[DF_tablenameType_IsActiveFlag] DEFAULT ((1)),
    [CreatedById] [int] NOT NULL,
    [CreatedDate] [datetime] NOT NULL,
    [ModifiedById] [int] NULL,
    [ModifiedDate] [datetime] NULL,
    CONSTRAINT [TablenameType_PK] PRIMARY KEY CLUSTERED
(
    [tablenameTypeId] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

An acceptable variation on this schema would be:

```
CREATE TABLE [dbo].[TablenameType] (
    [TablenameTypeId] [int] identity NOT NULL,
    [TablenameTypeDisplayShortName] [varchar](50) NOT NULL,
    [TablenameTypeDisplayName] [varchar](128) NOT NULL,
    [TablenameTypeDescription] [varchar](500) NOT NULL,
    [SortOrderNumber] [int] NOT NULL CONSTRAINT
[DF_tablenameType_SortOrderNumber] DEFAULT ((99)),
    [IsActiveFlag] [bit] NOT NULL CONSTRAINT
[DF_tablenameType_IsActiveFlag] DEFAULT ((1)),
    [CreatedByName] [int] NOT NULL,
    [CreatedDate] [datetime] NOT NULL,
    [ModifiedByName] [int] NULL,
    [ModifiedDate] [datetime] NULL,
    CONSTRAINT [TablenameType_PK] PRIMARY KEY CLUSTERED
(
    [TablenameTypeId] ASC
)WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

All column names are Pascal cased. This means that the first letter of each word is upper case followed by lower case letters up to the start of a new word. For example, if we need a lookup table for a Priority Code, the lookup table name will be **PriorityCodeType**.

Guidance on Designing Audit tables

Audit tables will be created using a similar strategy for each database. The strategy should ensure that future reference to the data and the transactions that brought the data to the current state can be tracked.

The strategy contains the following considerations:

1. Audit table names will be named in Pascal case with no underscore(s).
2. Audit tables will all start with the 'Audit' prefix ie "AuditPersonallInformation". Alternately it is acceptable to use a suffix of "Audit" ie "PersonallInformationAudit".
3. Audit tables will include a datetime stamp of the time the alteration was made and a user identifier of the user making the alteration.
4. Each row in audit tables will contain a code that will state if the transaction being logged is an:
 - a. Insert
 - b. Update
 - c. Delete
5. Audit tables will record all inserts, updates and deleted of the record.
6. Each audit record should include all the values whether changed or not, in the row of data that is being altered, with the metadata mentioned above.
7. All audit table column names other then the necessary metadata, AuditDatetime and AuditActionCode should come from the table of origin and replicate whatever names already exist.

Example:

Data Table:

```
CREATE TABLE [dbo].[TableData] (
    [TableId] [int] IDENTITY(1,1) NOT NULL,
    [TableName] [nchar](45) NOT NULL,
    [TableAlternateName] [nchar](45) NULL,
    [TableDescription] [nvarchar](max) NULL,
    [TableInitDate] [datetime] NOT NULL,
    [TableIsActiveFlag] [bit] NOT NULL,
    CONSTRAINT [tableData_PK] PRIMARY KEY CLUSTERED
    ([TableId] ASC)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
    [PRIMARY]
) ON [PRIMARY]
```

Audit Table:

```
CREATE TABLE [dbo].[AuditTableData] (
    [TableId] [int] IDENTITY(1,1) NOT NULL,
    [TableName] [nchar](45) NOT NULL,
    [TableAlternateName] [nchar](45) NULL,
```

```

[TableDescription] [nvarchar](max) NULL,
[TableInitDate] [datetime] NOT NULL,
[TableIsActiveFlag] [bit] NOT NULL,
[UserName] [nchar](25) NOT NULL,
[AuditDatetime] [datetime] NOT NULL,
[AuditActionCode] [nvarchar](7),
CONSTRAINT [auditTableData_PK] PRIMARY KEY CLUSTERED
([TableId] ASC)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
) ON [PRIMARY]

```

All column names are Pascal cased. This means that the first letter of each word is upper case followed by lower case letters up to the start of a new word. For example, if we need a audit table for proofing Hazardous Waste Reporter, the audit table name might be **AuditHazWasteReporter**.

Note that if the data table carries the creation date or insert date then the need for an “insert” record in the audit table is optional.

It is not anticipated that any existing audit table will be modified to meet this requirement. Existing tables are grandfathered in.

Date and Time Display Formats

Date fields will have a default format for data entry of MMDDYYYY, with no punctuation, or for historic data covering multiple centuries.

Date fields will have a format of MM/DD/YYYY for display only fields, or MM/DD/YY if space is a constraint.

Time fields will have a format of HHMM or HHMMSS, with no punctuation, for data entry.

Time fields will have a format of HH:MM or HH:MM:SS for display only fields.

Code Formats

Code fields can be either text or numeric dependant on use and design techniques. It is advisable to determine the initial format of the code field and then stay consistent with that format across all tables in the data collection. If possible it is recommended that the use of the domain of the particular code not

only remain consistent in the original system of definition but in all systems that use that code.